19 Database

Database

データベース（database, DB）とは、検索や蓄積が容易にできるよう整理された情報の集まり。データベースは物理設備です。

関係データベース（relational
database）は関係モデルにもとづいて設計、開発されるデータベースである。関係データベースを管理するデータベース管理システム (DBMS) を関係データベース管理システム (RDBMS) と呼ぶ。

Database Server

データベースサーバはリモートシステム。開発環境で同じPCにDBがインストールしてもリモート処理としてアクセスしています。「18 ネットワーク」章をご参照してください。

Database

Relations or tables

Table

Meta　データのカラム名称。

Data（Record）レコード部分

制約部

その他

SQL

DataType

よく利用するデータ型。

SQL and PL/SQL Data Type	ORACLE	MYSQL	POSTGRES	JDBC Mapping	説明
VARCHAR/VARCHAR2				java.lang.String	文字列
NCHAR/NVARCHAR2				oracle.sql.NString	文字列
TEXT				java.lang.String	文字列
NCLOB				oracle.sql.NCLOB	文字列
SMALLINT				int	整数
INTEGER				int	整数
DEC, DECIMAL, NUMBER, NUMERIC				java.math.BigDecimal	数値
DOUBLE PRECISION, FLOAT				double	数値
REAL				float	数値
DATE				java.sql.Timestamp	日付
TIMESTAMP				java.sql.Timestamp	日時
BOOLEAN				boolean	論理
CLOB				java.sql.Clob	CLOB
BLOB				java.sql.Blob	バイナリデータ

Create

The CREATE TABLE statement is used to create a new table in a database.

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
    PRIMARY KEY (column1, column2)
);
```

例：

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    PRIMARY KEY (ID)
);
```

レコードを作成します。

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

例：

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

Read

Query章をご覧ください。

Update

The UPDATE statement is used to modify the existing records in a table.

```
UPDATE table_name
SET column1 = value1, column2 = value2, ...
WHERE condition;
```

例：

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
```

Delete

Databaseを削除

```
DROP DATABASE databasename;
```

Tableを削除

```
DROP TABLE table_name;
```

Recordを削除

The following SQL statement deletes the customer "Alfreds Futterkiste" from the "Customers" table:

```
DELETE FROM table_name WHERE condition;
```

例：

```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

Alter (Postgres)

Table名変更

カラム名称変更

カラム名変更

カラムを削除

Query

検索元のTBL：Employee に以下データ（レコード）が収納されています。

ID	NAME	AGE	SEX
1	山田	20	1
2	山田	30	1
3	佐藤　一郎	30	0
4	佐藤　次郎	20	0

SELECT

The SELECT statement is used to select data from a database.The data returned is stored in a result table, called the result-set.

```
SELECT column1, column2, ...
FROM table_name;
```

例：

```
SELECT * FROM employee;
```

取得した結果は上記TBLのすべてレコード。

「*」表記

取得した項目のエリアに記載すると、取得元のすべてカラムを抽出します。*ではない具体的カラムを記載した場合、指定されたカラムのみ抽出します。

例：

```
SELECT ID, NAME FROM EMPLOYEE;
```

取得して結果は青い背景の部分。

ID	NAME	AGE	SEX
1	山田	20	1
2	山田	30	1
3	佐藤　一郎	30	0
4	佐藤　次郎	20	0

WHERE

Whereは絞り条件と呼ばれています。TBL全件データから指定された条件を絞って、レコードをフィルター(Filter)します。

例１：

```
SELECT
 *
FROM
EMPLOYEE
WHERE
NAME = '山田'
;
```

ID	NAME	AGE	SEX
1	山田	20	1
2	山田	30	1
3	佐藤　一郎	30	0
4	佐藤　次郎	20	0

例２：

```
SELECT * FROM EMPLOYEE WHERE AGE < 30
```

取得して結果は

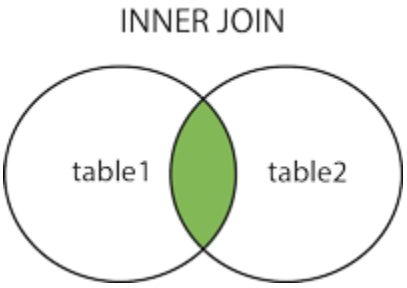
ID	NAME	AGE	SEX
1	山田	20	1
2	山田	30	1
3	佐藤　一郎	30	0
4	佐藤　次郎	20	0

JOIN

INNER JOIN

The INNER JOIN keyword selects records that have matching values in both tables.

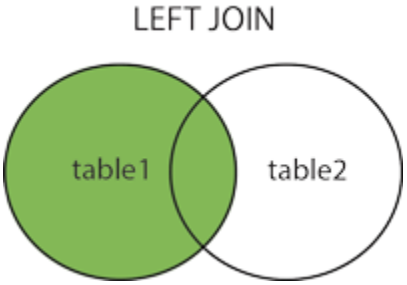
```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```



LEFT JOIN

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

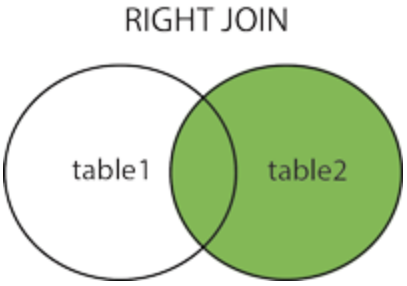
```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```



RIGHT JOIN

The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

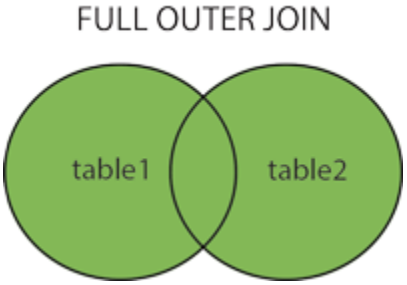
```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```



FULL OUTER JOIN

The FULL OUTER JOIN keyword return all records when there is a match in left (table1) or right (table2) table records.

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```



> FULL OUTER JOIN and FULL JOIN are the same.

SubQuery

```
Select
  NAME
  , LOCATION
  , PHONE_NUMBER
from
  DATABASE
WHERE
  ROLL_NO IN (SELECT ROLL_NO from STUDENT where SECTION = 'A');
```

Alias

Order By

ASC

DESC

検索の基本考え方

1. データ取得元（何処からデータを取得する）
2. どのようなカラムを取得すべき（何を取得する）
3. 絞り条件は？
4. 結合が利用する為：Join先、Join元、FKは
5. ソート条件

Transcation

DB設計

A5Mk2

第一正規型

第二正規型

第三正規型

設計書

基本DB設計書は一番大切な資産であること！基本DB設計書は以下種類があります。

1. ER図
2. Table定義書
3. CURD図

ER図

ERとはEntity Relationship。Tableの関係を表すものです。Relationは以下３種類関係があります。

one to one

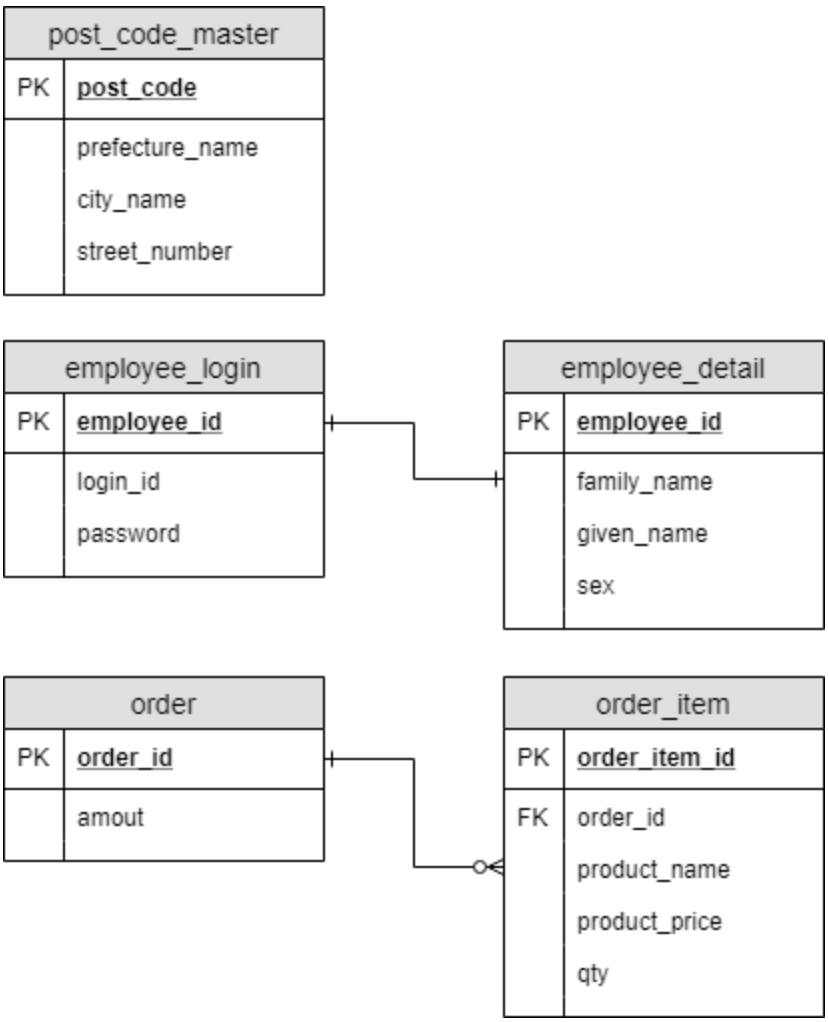
例：社員ログイン（employee_login）、社員詳細（employee_detail）の関係は１： １です。

one to many

例： １つのオーダーについて、複数の商品注文できます。オーダー（order）とオーダーアイテム（order_item）は１：N関係です。

many to one

one to many 関係の逆転です。マインTBLの立場が違うのみです。



Table定義書

CRUD図

> ER図を印刷して手元にもつは開発者の常識です。できないなら別の業界を転職してください。

SQL補足

COUNT

SUM

TRIM

CONTACT

カラムを取得する時、文字列を連結する操作ができます。

```
SELECT
  family_name || ' ' || AS full_name given_name
from
  Employee;
```

DATE

現在の日時の取得：

```
INSERT INTO EMPLOYEE VALUES (1, '山田', NOW());
```

※関数はDB依存します。ORACLEの場合、SYSDATEを利用してください。

DATEと文字列の変換：

NOT EXISTS

The EXISTS operator is used to test for the existence of any record in a subquery.

```
SELECT column_name(s)
FROM table_name
WHERE EXISTS
(SELECT column_name FROM table_name WHERE condition);
```

CASE WHEN

The CASE statement goes through conditions and returns a value when the first condition is met (like an IF-THEN-ELSE statement). So, once a condition is true, it will stop reading and return the result. If no conditions are true, it returns the value in the ELSE clause.If there is no ELSE part and no conditions are true, it returns NULL.

```
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    WHEN conditionN THEN resultN
    ELSE result
END;
```

SELECT INSERT

The INSERT INTO SELECT statement copies data from one table and inserts it into another table.

```
INSERT INTO table2
SELECT * FROM table1
WHERE condition;
```

ROW_NUMBER

読み方の差異

SQL／DATABASE／設計する時、同じものに対して複数の読み方があります。以下表ご参照してください。

SQL	データベース定義	備考
Row	Record：レコード	レコード：データ部
Column	Attribute/Field：属性／フィールド	カラム／属性：Meta部、項目の定義
Table	Relation：リレーション	Entity：エンティティ
Sequence		Sequence：シーケンス

※単語を正しく理解してください。

質問